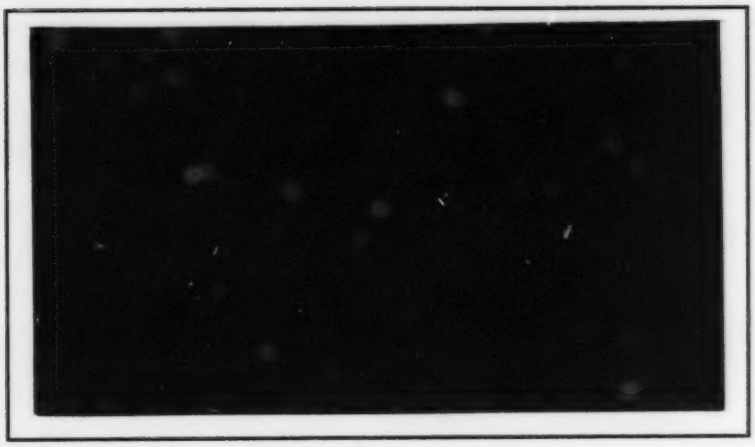

CENTRE FOR
SYSTEMS
SCIENCE

LCCR

Simon Fraser University, Burnaby, B.C., Canada V5A 1S6 / (604) 291-4704

Reaching Byzantine Agreement in Amortized Constant Number Rounds

Xiaotie Deng and Tiko Kameda

CSS/LCCR TR 91-04

June 1991

Reaching Byzantine Agreement in Amortized Constant Number of Rounds*

Xiaotie Deng† and Tiko Kameda†

January 28, 1991

(Extended Abstract)

Abstract. *Consider the Byzantine Agreement (BA) Problem among n generals, up to t of whom may be traitorous. It is well-known that agreement is possible if and only if $n > 3t$ and that the number of rounds required is at least $t + 1$. Suppose it is needed to reach BA m times for some integer m. We are interested in minimizing the total number of rounds in reaching BA m times, for a large m, improving the obvious bound of $O(mt)$. In this paper, we present a protocol which executes m instances of Eventual Byzantine Agreement within $O(t^2 + m)$ rounds, which is linear in m when $m = \Omega(t^2)$.*

Since reaching BA is much simplified if the commander is loyal, our goal is to select a loyal general as the commander during the course of repeated agreements, if the original commander is not loyal; from that time on, others can simply follow the commander, using only two rounds per instance. If $n \leq 3t$, it is, of course, impossible to achieve that goal. For the case $n > 3t$, our approach is elect a (commanding) "council" by removing a group of at most three generals, at least one of whom is traitorous. By repeating this, smaller and smaller councils can be elected, from which the successive new commanders are selected.

The complete algorithm implementing this idea is obtained by a novel combination of various versions of agreements: Eventual Byzantine Agreement [DRS], Interactive Consistency [PLS], and Byzantine Firing Squad [BL,CDDS].

---

*This work was partially supported by the NSERC (Natural Sciences and Engineering Research Council of Canada) International Postdoctoral Fellowship and research grants from the NSERC and the Advanced Systems Institute of British Columbia.

†School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6, Netaddress: {deng,tiko}@cs.sfu.ca

92-05830/

1 Introduction

Byzantine Agreement (BA) is one of the most widely studied problems in distributed computing. It is well known that if we require that processors reach an agreement in the same round, at least $t+1$ rounds are needed [DRS, FL], where t is an upper bound on the number of faulty processors. Randomization, however, can be used to overcome this lower bound. There are randomized protocols achieving BA in expected constant time [DSS, FM]. This gives the impression that randomized protocols are more powerful for BA than deterministic ones. In this paper, however, we would like to study whether, when we need to perform BA repeatedly, this constant number of rounds can also be achieved by a deterministic protocol, on the average per BA instance.

The answer is "No" for Immediate Byzantine Agreement (IBA), in which processors are required to reach an agreement in the same round. A lower bound of $t+1$ rounds per instance is proven for all IBA protocols [DRS, FL]. Even when faulty processors do not cheat and behave as if they were non-faulty in a particular instance of BA, protocols must run for $t+1$ rounds. Thus, it is impossible to decrease the number of rounds from $\Omega(mt)$ for m instances of IBA.

In *Eventual Byzantine Agreement (EBA)*, a weaker notion of agreement is used, and it is not required that all processors come to an agreement in the same round, as long as they eventually do. In some distributed database applications, for example, EBA may be quite adequate [DRS]: once a site knows the final value to be decided upon, it can write that value before other sites make up their minds. Dolev, et al. [DRS], show that it is possible to stop earlier in EBA than in IBA.

In this paper, we design a protocol achieving an amortized constant number of rounds for repeated EBAs. One difficulty in designing such a protocol is that faulty processors are difficult to identify. Faulty processors can behave in such a way that some non-faulty processors cannot tell for sure which processors are faulty. In this paper, therefore, we choose, instead of finding faulty processors, to design a protocol which will elect a smaller and smaller council for making BAs on behalf of the others.

Another difficulty arises when we try to design a protocol for excluding faulty processors from the decision making council. They may cheat in such a way that after many rounds, the processors end up with the previous council, which will increase the communication complexity without accomplishing anything. Fortunately, there is a way to overcome this difficulty.

In such a protocol, all the non-faulty processors need to agree on the same council. In particular, all the non-faulty council members must know the council which is used in a particular round, especially in the case when the executions of two successive BAs overlap. Our method used in resolving this difficulty is very similar to the *Byzantine Firing Squad* (BFS) protocol of [BL].

As in [PSL], our model makes the following assumptions:

- Communication is point-to-point and completely reliable.
- The receiver of a message can determine its sender.
- The protocol is synchronous, i.e., it proceeds in rounds. In each round, every general first sends as many messages as he wishes to other generals, and then receives the messages sent to him by others.

We now formally introduce the concepts used in this paper. An arbitrary failure of a processor, which represents any deviation from its specification, is abstracted as the *Byzantine failure* [PSL]. Consider n generals (processors), one of whom is the *commander* and the rest *lieutenants*. Up to t generals, including the commander, may be *traitorous*, i.e., subject to Byzantine failures. A general that always behaves correctly is said to be *loyal*. The commander tries to broadcast a "value" (in the simplest case, 0 or 1, which can be interpreted as "Advance" or "Retreat") to his lieutenants. *BA* is said to be reached if 1) any two loyal lieutenants decide on the same value, and 2) if the commander is loyal, every loyal lieutenant decides on the commander's value.

In *IBA*, all the loyal generals must reach an agreement in the same round, knowing that all other loyal generals will do the same [DRS]. In *EBA*, on the other hand, it is not required that all generals come to an agreement in the same round, as long as they eventually do. The *Interactive Consistency* (IC) problem is similar to the BA problem, except that there is no commander and each general G_i has an initial value v_i. The goal of an IC protocol is for all the loyal generals to agree on a single *consensus vector* such that the components of the vector corresponding to the loyal generals are their private values. In a *BFS* protocol, when a START signal is received by a loyal general, he initiates the synchronization activity, so that all the loyal generals subsequently reach a distinct *Firing* state in the same round. Any protocol for BFS, by definition, must satisfy the following conditions. 1) *Agreement*: If any loyal general fires in some round, then all loyal generals fire in that round. 2) *Validity*: If any loyal general receives a START signal, then some loyal general eventually fires.

In Section 2, we introduce a protocol for electing a new council for the simplest case where $t = 1$, i.e., there is at most one traitorous general, in order to give a general idea behind the protocol for finding a loyal commander. Section 3 generalizes our council election protocol to arbitrary t. In Section 4, we combine the council election protocol with an EBA and BFS protocol to obtain a complete solution for repeated instances of BA, and prove correctness. Section 5 concludes the paper with some remarks and open problems.

2 Partial Solution for the Case $t = 1$

To illustrate the difficulties involved in our problem and techniques used to resolve them, we present, in this section, a protocol for the simplest non-trivial case where there are four generals with at most one traitorous general among them, i.e., $n = 4$ and $t = 1$. Obviously, repeated BAs can be accomplished in the number of rounds linear in the number of BA instances in this simple case. We will therefore focus on a council election protocol.

Let G_0 denote the commander and G_i, $i = 1, 2, 3$ the three lieutenants. Our protocol consists of 3 stages with a total of 4 rounds. Informally, it proceeds as follows. In the first stage (consisting of the first two rounds), the generals execute one instance of BA to agree on a common value. In the second stage (the third round), those lieutenants who got conflicting messages in the first stage each propose (a council consisting of) a new commander. In the third stage (the 4th round), the generals elect one general as the new commander out of the candidates proposed in the second stage by choosing the candidate with the minimum ID.

We now describe our protocol in more detail. In the first stage, we use the protocol given in [PSL]. In the second and third stage, three concurrent instances of the protocol in [PSL] are used, one for each lieutenant considered as the commander. Each round is described in 2 steps; the sending step and the receiving step.

Election Protocol

Round 1:
- The commander G_0 sends out his value,
- which are received by the three lieutenants G_i, $i = 1, 2, 3$.

Round 2:
- Each lieutenant relays the value received from G_0 in Round 1 to all the other lieutenants.[1]
- Each lieutenant decides his value by the majority of all the values he received (one directly from G_0 in Round 1, the others relayed by the other generals in Round 2). If no two received values agree or the majority is the NULL message, he decides on the unique default value.

Round 3:
- If the value received by a lieutenant from G_0 in Round 1 does not agree with the value relayed by another lieutenant, he selects the remaining lieutenant (i.e., not himself) as a candidate for the new commander, and he broadcasts his proposal to all others.

[1] If no message is received from G_0, he pretends as if the NULL message were received.

Round 4:

- Each general that received proposals in the previous round relays them to all other generals.[2]
- Each general accepts or rejects a proposal by the majority (two out of three) of the messages carrying it (one directly from the proposer and two relayed ones). He then elects as the next commander the candidate with the minimum ID from among those accepted proposals.

In the above description, all the messages deviating from the protocol are ignored.

If a lieutenant receives a value v from the commander G_0 in Round 1, and receives a value different from v from another general G_i in Round 2, he knows that either G_0 or G_i must be traitorous and can propose, in Round 3, the remaining lieutenant as the new commander. Note that, since no general can propose himself as the new commander, all the candidates, even the one proposed by the traitorous general in Round 3, are loyal. Therefore, all we need to show now is that all loyal generals will agree on the same set of candidates in Round 4. One can look at this problem as the IC problem with new commander proposals as private values of individual generals. The details can be worked out easily and are omitted. Thus, we have

Theorem 1. Suppose there are four generals and at most one of them is traitorous. Once any general deviates from the BA protocol, a loyal commander will be elected by the election protocol. □

One serious problem with the above Election Protocol is that the candidate set can be empty. That is, it is possible that after an instance of BA, we execute two extra rounds in vain, i.e., without reducing the size of the council. If this situation persists every time, we will have used $4m$ rounds for m instances of BA, which is worse than the straightforward method of repeating 2 $(= t + 1)$ rounds per instance. In the next section, we will address and solve this problem, by guaranteeing that, after each instance,

- if no valid candidate is proposed, the election process is terminated after a constant number of extra rounds, and
- if a valid candidate is proposed, the size of the council is reduced by 3.

3 Council Election for the General Case

When there are several traitorous generals, there are some complications that we must cope with in order to achieve an amortized constant number of rounds.

[2] Again, if no message is received from a certain general, take NULL as his message.

Informally, the approach we will take is as follows. Instead of always invoking a BA protocol involving all generals, a council of generals is elected, which first makes a decision and broadcasts it to the remaining generals. The latter will simply accept the decision of the council (say, by the majority vote on the DECISION messages sent out by the council members). In the first of the m instances of BA, the council consists of all generals. Whenever it is suspected that there is a traitorous general in the current council, we want to elect a new council by removing three generals from it, at least one of whom is traitorous. If this can be somehow accomplished, the new council will always contain less than 1/3 traitorous generals.

Two immediate questions in implementing the above idea are 1) how to detect that the current council contains a traitorous general? 2) how to make sure that at least one traitorous general is removed every time a new council is formed?

Definition 1: A traitorous general is said to exhibit a *consequential erroneous behavior* if (a) he incorrectly relays a message (including the NULL message) from a loyal general to another loyal general, or (b) he originates contradictory messages to two or more loyal generals in a round. □

Since the definition of consequential erroneous behavior involves a loyal general, if G is a loyal general who receives a message M from general A in a certain round and M' (claimed to be the message received from A in the previous round) from general B, then G can conclude that either A sent contradictory messages to B and G in the previous round or B incorrectly relayed the message from A or both. In any case, G can detect that some traitorous general has exhibited a consequential erroneous behavior. Thus, we have

Lemma 1: Any consequential erroneous behavior by a traitorous general can be detected by a loyal general. □

We call the comparison of the messages, one from its originator and the other from a relayer, the *consistency check*. The above lemma shows that the consistency check by a loyal general can always identify a set of two generals which contains the offending traitorous general ($\{A, B\}$ in the paragraph just before Lemma 1.) Therefore, if G proposes a new council excluding A, B and G (himself), then the proposed council will contain at least one fewer traitorous general than the current council. We make the proposer exclude himself from the proposed council, so that a traitorous proposer has no choice but to exclude himself from the new council that he proposes. Based on the above discussion, we adopt the following Council Proposal (CP) Rule:

CP Rule: If a message from a general A and its relayed version from general B don't agree, propose a new council excluding A, B, and the proposer. □

The problem now is how to design a protocol for the loyal generals to use in order to agree

on the same new council once a proposal is made by some (loyal or traitorous) general. At first glance, this resembles the *IC* problem: if a general proposes a new council, his private value is the council he proposes. If a general does not propose a council, his private value is considered as NIL. Then an IC protocol can be invoked, with values defined as above, to construct the same consensus vector of councils for all the loyal generals. Using some appropriately predefined total order among the councils, the loyal generals can then pick the "minimum" council in the vector. Similarly, they can elect the new commander as the "minimum" general in the new council.

Unfortunately, as pointed out at the end of Section 2, there is a hitch in the above approach: What if NIL is elected as the new council? To avoid this situation, we need to make sure that at least one loyal general makes a proposal for a new council whenever the election protocol is invoked.

The next question that we need to address is: When and how should the generals invoke the IC protocol to elect a new council? Since the IC protocol requires that all loyal generals initiate it at the same time, the problem we need to solve is similar to the BFS problem [BL,CDDS]. In fact, our requirements are:

(a) If any loyal general receives a proposal to elect a new council, then in some future round, all the loyal generals will "fire," i.e., simultaneously start the IC protocol for the new council election.

(b) There is at least one loyal general with a non-NIL new council as its private value when the generals "fire," i.e., at the initiation of the IC protocol.

Notice that point (a) is the standard requirement for the BFS problem, but point (b) is a special requirement for our problem. Thus, we shall use the BFS protocol given in [BL], with a modification to accommodate the special requirement (b) given above.

Protocol CSS (Council Selection Synchronization)
while not having made a council proposal in the current instance of BA **and** the current round is not the firing round of BFS for the current instance of BA **do** in each round
{Perform consistency check on all messages received;

- **if** Rule CP is applicable, then apply it to make a proposal, and relay the proposal to all other generals. If the BFS protocol has not been initiated in the current instance of BA, initiate the BFS protocol; otherwise, follow the BFS protocol.

- **else if** a "valid" proposal is received from some other general, relay it to all other generals (where a proposal is valid if it excludes three generals, including the sender, from the current council). At the same time, follow the BFS protocol.

}

At the firing round of BFS, use the proposal, if any, as the private value; use NIL otherwise. □

Lemma 2: In protocol CSS, if BFS is initiated, then by the time the firing round of the BFS is reached, either at least one loyal general has a non-NIL proposal or there is a traitorous general who has behaved like a loyal general by making the same valid proposal to all the other generals.

Proof Sketch. The lemma is already true if a loyal general makes a council proposal on his own in a round to initiate the BFS. Suppose, therefore, that the BFS is initiated by a traitorous general, who makes the first valid council proposal to one or more loyal generals and no loyal general initiates a proposal on his own. Consider any loyal general G. If G receives the same proposal relayed from all other generals as a result of their carrying out the **else if** part of protocol CSS, then the proposal must have sent the same proposal to all the other generals. If not, G will follow Rule CP in the third round and make his own proposal. □

By Lemma 2, once the BFS protocol is initiated, at least one component in the vector of councils will be non-NIL, and a new council will be elected in $2(t+2)$ rounds after the BFS is initiated ($t+3$ [BL] rounds for the BFS protocol and $t+1$ rounds for the IC protocol).

4 A Protocol for Repeated Byzantine Agreements

We are now in a position to put the pieces presented so far together and state a protocol for repeated BAs.

Protocol RBA

Initially *COUNCIL_NO* = 1 and *COUNCIL(1)* = {all generals}.

Algorithm NCM (for a non-council member)

NCM1: Wait until the same *DECISION* is received from the majority of *COUNCIL(COUNCIL_NO)*. Adopt the majority decision value.

NCM2: When the majority of the current council members send the same new council in the same round, increment *COUNCIL_NO*, and record *COUNCIL(COUNCIL_NO)*.

NCM3: Remain to be a non-council member.

Algorithm CM (for a council member)

CM1: To initiate a new instance of BA, the commander sends out his command in the first round to all the members of *COUNCIL(COUNCIL_NO)*.

CM2: Each council member follows an EBA protocol until an eventual agreement is reached.

CM3: Concurrently, run protocol CSS given in the previous section.

CM4: In the firing round, start the IC protocol to elect a new council, with the decided proposal from the last step. If he tries to send a different value in the IC protocol, that value will be ignored.

CM5: When the IC protocol ends with a vector of private values, choose the "min" non-NIL value, and choose the unique "min" commander. Send the decision out to all the non-council members. The current council members not included in the new council become noncouncil members. Increment *COUNCIL_NO*.

The above protocol gives us the answer to the main question for repeated BAs:

Theorem 2. Among n generals with at most t traitorous ($n > 3t$), k repeated EBAs can be reached in $O(t \cdot min(k,t) + k)$ rounds.

Proof Sketch. First, we assume that time interval between two successive BAs is so long that all the activities in our protocol have ended before the next instance of BA is required. The correctness depends on the following two facts:

B1. The condition $n' > 3t'$ is satisfied for all the councils elected, where n' is the size of the council and t' is an upper bound on the number of traitorous council members.

B2. The majority of the decisions of council members gives the agreed-upon value of the loyal generals in the council.

Fact *B1* guarantees that BA can be reached successfully. It is true since every new council excludes at most three incumbent council members, of whom at least one is traitorous. Fact *B2* follows from Fact *B1* and properties of BA. To prove the correctness of the council election protocol, the following conditions need to be verified.

C1. The condition $n' > 3t'$ is satisfied for all the councils elected.

C2. When the IC protocol is initiated, every loyal general will participate.

C3. A new council with a new commander is agreed upon by all the loyal generals at the end of the IC protocol.

The condition *C1* is the same as *B1* and is satisfied at the initiation of the protocol. Thus, it will be always satisfied if new councils are proposed as described. Condition *C2* is guaranteed by the protocol for synchronizing council election. Condition *C3* follows from Lemma 2 and the requirement in Step CM4 that forbids a traitorous general from changing the proposal during the IC protocol from what he sent out while initiating the synchronizing protocol. In Step CM5, all the loyal generals will have the same vector of valid councils, according to the property of IC. If a loyal general has a non-NIL proposal, then obviously, the vector has at least one non-NIL proposal. Otherwise, from Lemma 2, there is a traitorous general who sent the same valid council to all the generals in the first round. That new council will be the value in the vector for that traitorous general. Thus, the vector always has at least one non-NIL component.

Finally, we need to guarantee that the three subprotocols, an EBA protocol, a DFS protocol, and an IC protocol, are correctly linked together. If no general cheats in the first two rounds of EBA, the loyal generals can obtain the value of BA in a constant number of rounds [DRS]. Otherwise, protocol CSS will be initiated. As proven above, that will finally lead to a smaller new council for the subsequent instances of EBAs. There are two possibilities in the execution of the protocol: 1) EBA is reached in a constant number of rounds, and protocol CSS is not initiated; 2) protocol CSS is initiated.

The first case always adds only a bounded number of count to the number of rounds during each EBA instance. For the second case, a new council containing at least one fewer traitorous general will be elected. Thus, the second case only occurs at most t times. The count for k repeated EBAs in the theorem follows immediately from the above facts.

If we drop the assumption that two successive instances of BA never overlap so that our protocol becomes quiescent in between, we need to know when and how often BAs are needed. Suppose one instance is needed at predetermined fixed rounds, then the above analysis carries over immediately. This is because our BFS and IC protocols guarantee that the new council is agreed upon at the same round by all the loyal generals. Otherwise, i.e., if the commander can start an instance of BA in an arbitrary round, we must assume that any council member can start a BA instance by faking a relay message from the commander, because the commander can be traitorous and can send a start message to some generals while ignoring others. In that case, we need to modify our EBA protocol in such a way that other council members may ask the commander about his value, if someone relayed a message of the commander but he didn't get one from the commander directly. The modification is straightforward and is omitted. □

5 Open Problems and Discussions

Ultimately, we would like, if possible, to obtain a protocol which reaches k repeated EBAs with $O(t+k)$ rounds of message exchanges among the processors in the system, where t is the number of faulty processors. The protocol presented in this paper is made up of protocols for three similar problems: Eventual Byzantine Agreement, Byzantine Firing Squad, and Interactive Consistency. One may notice that, our subprotocol for Interactive Consistency achieves an immediate rather than eventual agreement. A natural question is: Is it possible to achieve EBA for the combined requirement of Byzantine agreement and council election so as to either elect a new council or stop after a constant number of rounds? Another question related to the main concern of this paper is whether we can design a protocol in such a way that, if f processors actually exhibit faulty behavior, the protocol will select a new council which excludes all f of them, and unavoidably, as we have seen, together with some innocent nonfaulty processors but no more than $2f$ of them. These are, probably, the two most important questions to be answered in order to prove/disprove the above conjecture.

References

[BL] J. Burns and N.A. Lynch, "The Byzantine firing squad problem," MIT Technical Report MIT/LCS/TM-275, 1985.

[CDDS] B. Coan, D. Dolev, C. Dwork, and L. Stockmeyer, "The distributed firing squad problem," *Proc. 17th STOC*, 1985, pp.335-345.

[DSS] C. Dwork, D. Shmoys, and L. Stockmeyer, "Flipping persuasively in constant expected time," *Proc. 27 FOCS 1986.*

[DRS] D. Dolev, R. Reischuk, and H.R. Strong, "Eventual is earlier than immediate," *Proc. 23th FOCS* 1982, pp.196-203.

[FL] M.J. Fischer and N.A. Lynch, "A lower bound for the time to assure interactive consistency," *Inform. Proc. Letts.*, 14:4, 1982, pp.183-186.

[FM] M.J. Fischer and N.A. Lynch, "Optimal algorithms for Byzantine Agreement," *Proc. 20th STOC 1988*, pp.148-161.

[PSL] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM 27*(1980), pp.228-234.

Centre for Systems Science

• Simon Fraser University established the **Centre for Systems Science** as a provincial centre for advanced study. The objective of CSS is twofold: coordinate and combine advanced systems research at SFU, and encourage application of research results within industry in order to further the technological development of British Columbia.

• CSS has assembled three research groups to streamline its efforts. The orientation of these groups is as follows: intelligent systems, microelectronics, and computer and communications systems. Because of shared research interests, this latter group merges with LCCR (**Laboratory for Computer and Communications Research**) for the publication of technical reports.

• CSS's three groups involve over 80 skilled faculty researchers principally from SFU's Schools of Engineering Science and Computing Science, but also draw substantially on Mathematics, Physics, Geography, Philosophy, Linguistics, Psychology, Communication, and Kinesiology.